MINI PROJECT

REPORT ON

"MAKE WEALTH"

Submitted By
"Anetta Jose, Bagul Devendra, Kumari Khushboo, Barvi Tanmay, Mankar Kanchan "

Under the Guidance of
Prof. Mrs Leena Patil

Submitted To
Savitribai Phule Pune University [SPPU]

As a partial fulfillment for the award of the degree of
MASTER OF COMPUTER APPLICATION Semester : 1

At
ASM's
Institute of Business Management and Research, Chinchwad, Pune – 19
(Affiliated to SPPU & Approved by AICTE)

Session: 2021-23

INDEX

INTRODUCTION

The banking sector is the lifeline of any modern economy. It is one of the important financial pillars of the financial sector, which plays a vital role in the functioning of an economy. It is very important for the economic development of a country that its financing requirements of trade, industry, and agriculture are met with a higher degree of commitment and responsibility. Thus, the development of a country is integrally linked with the development of banking. In a modern economy, banks are to be considered not as dealers in money but as the leaders of development. They play an important role in the mobilization of deposits and disbursement of credit to various sectors of the economy.

The current process of transformation should be viewed as an opportunity to convert banking into a sound, strong and vibrant system capable of playing its role efficiently and effectively on its own without imposing any burden on the government.

This MakeWealth banking is a software application for maintaining a person’s account in a bank. This project is a client-server-based desktop application system, to computerize at bank work.

MakeWealth Banking is an application for maintaining a person’s account in a bank. This project currently includes modules like OpenAccount, Edit account or Close account, Monthly statement, withdraw money, Bank summary, Reports, etc.

SCOPE OF SYSTEM

In the future, more software companies will hire this software program because nowadays the need for speed in day-to-day life has become essential. As competition increases, companies by considering old versions, develop more efficient versions for individual success.

In the future, we can make a link of this project with networking. We can also convert this project to Hibernate. Hibernate is a framework. It stores a large number of databases. In the future, we can also add the facility to change the password and recover the forgotten password.

PROPOSE SYSTEM

The main purpose of implementing the MakeWealth Banking system is to help banking staff to do customers' banking work easily with greater accuracy and efficiency. To provide an efficient, effective, and secure way of the system between user and customer. To strengthen customer relations with the Banks. MakeWealth offers personalization and scalability in maintaining user engagement. The proposed system can be used to maintain the Banking schedule. In larger organizations employees are large.

- The main purpose of implementing the Make Wealth Banking system is to help banking staff to do customer’s banking work easily.
  - Personalization
  - Scalability
  - To maintain Banking schedule
  - Accuracy
  - Efficiency

ADVANTAGES OF PROPOSED SYSTEM

- User Authorized system
- Reduces Clerical work
- Great Speed
- Time Consuming
- User Friendly
- Enable user’s workspace

EXISTING SYSTEM

The existing system works manually. The existing system has got a lot of intricacies within itself and needs a lot of human effort and paperwork. All the data needs to be maintained on ledgers and maintaining this is a tedious and risky process. As the transactions increase, so does the data too. So the task of maintaining them increases exponentially. To view data may need a lot of paper to be searched. Some of the negative aspects of the existing systems are:

1) Time Consuming: There is a lot of time consumed in the bank, because of software issues which are not maintained properly.

2) Reliability: This banking system is not fully reliable whenever the computer system creates a problem and does not work properly then sometimes our data is damaged or lost.

3) Man Power: In this project, manpower is fully used. Some employees need to manage the banking system.

4) Less Accurate: This system is not fully accurate, because sometimes a computerized system creates problems in working, then the computer system also gives us wrong results. To overcome these, we have come up with our system.

FEASIBILITY STUDY

Feasibility study is to check the availability of the project under consideration. Theoretically various types of feasibilities are conducted but we have conducted three types of feasibilities explained roundly.

- **Economic Feasibility**

With the manual system the operating cost of the system is about 20, 00,000 P.A. This cost comprises a salary of 2 person, stationary, various files that contains numerous patient records and also a large storage space to store all these data but the Make Wealth Bank System every bit of information is computerized which reduces the cost of stationeries like papers for writing customer records, staff record, transaction history etc. And also reduces large storage and holds large amount of data. Hence the system is economically feasible reducing the manual cost to about 5, 00,000 P.A. Due to the use of limited amount of resources the system is Eco-Feasible.

- **Technical Feasibility**

The Make Wealth Bank System is a user-friendly and easy to operate with all the create, delete, update and search commands, Make Wealth Bank System is the latest system which is very easy to handle. The new system can be handled by a 3 operators (admin) , Staff and Customers. Admin are sufficient since our existing system is computerized it produce the desired output in given time any piece of information can be searched from a large amount of data at a particular speed and it also processes a certain volume of data in response time. We use Java. The designing of front-end of any project is very important so we select Java as the front end due to the following reasons

- Simple and Familiar
- Compiled and Interpreted with Platform Independent
- Portable, Robust, Secure
- Object-Oriented
- Distributed
- Multi-threaded and Interactive
- High Performance
- Dynamic and Extensible

We have selected MYSQL as our back end because of the following reason

- Able to handle large data
- Easy way of storing large data
- Easy way of maintaining data-(inserting, updating, searching, deleting)
- Data Management

- **Operational Feasibility**

  The new system is feasible in all senses. The new system meets the emerging needs of the current situation providing large storage and holding a bulk amount of data in a system format which makes data handling easier. At present stage all work is done electrically so throughput and response time is less.

SURVEY OF TECHNOLOGIES

CODE EDITORS:

Eclipse

Eclipse IDE is written in Java. It mainly consists of a base ‘Workspace’ and a plug-in system so that we can add more features to it through plug-in and extend the functionality of the IDE.

Features :

- Supports various source knowledge tools like folding and hyperlink navigation, grading, macro definition browser, code editing with syntax highlighting.
- Provides excellent visual code debugging tool to debug the code.
- Eclipse has a wonderful user interface with a drag and drop facility for UI designing.

LANGUAGES:

Java

Java is a programming language and a platform. Java is a high-level, robust, object-oriented, and secure programming language.

Features :

- Simple and Familiar
- Compiled and Interpreted with Platform Independent
- Portable, Robust, Secure
- Object-Oriented

- Distributed
- Multi-threaded and Interactive
- High Performance
- Dynamic and Extensible

DATABASE:

MySQL

MySQL is a widely used relational database management system (RDBMS). MySQL is free and open-source. MySQL is ideal for both small and large applications.

Features :

- Open-Source
- Quick and Reliable
- Scalable
- Secure
- Support for large databases

SOFTWARE AND HARDWARE REQUIREMENTS

Software Requirement	
Operating System	Microsoft Windows
Software	
Front-End Software	JAVA
Back-End Software	MYSQL
Server	XAMPP

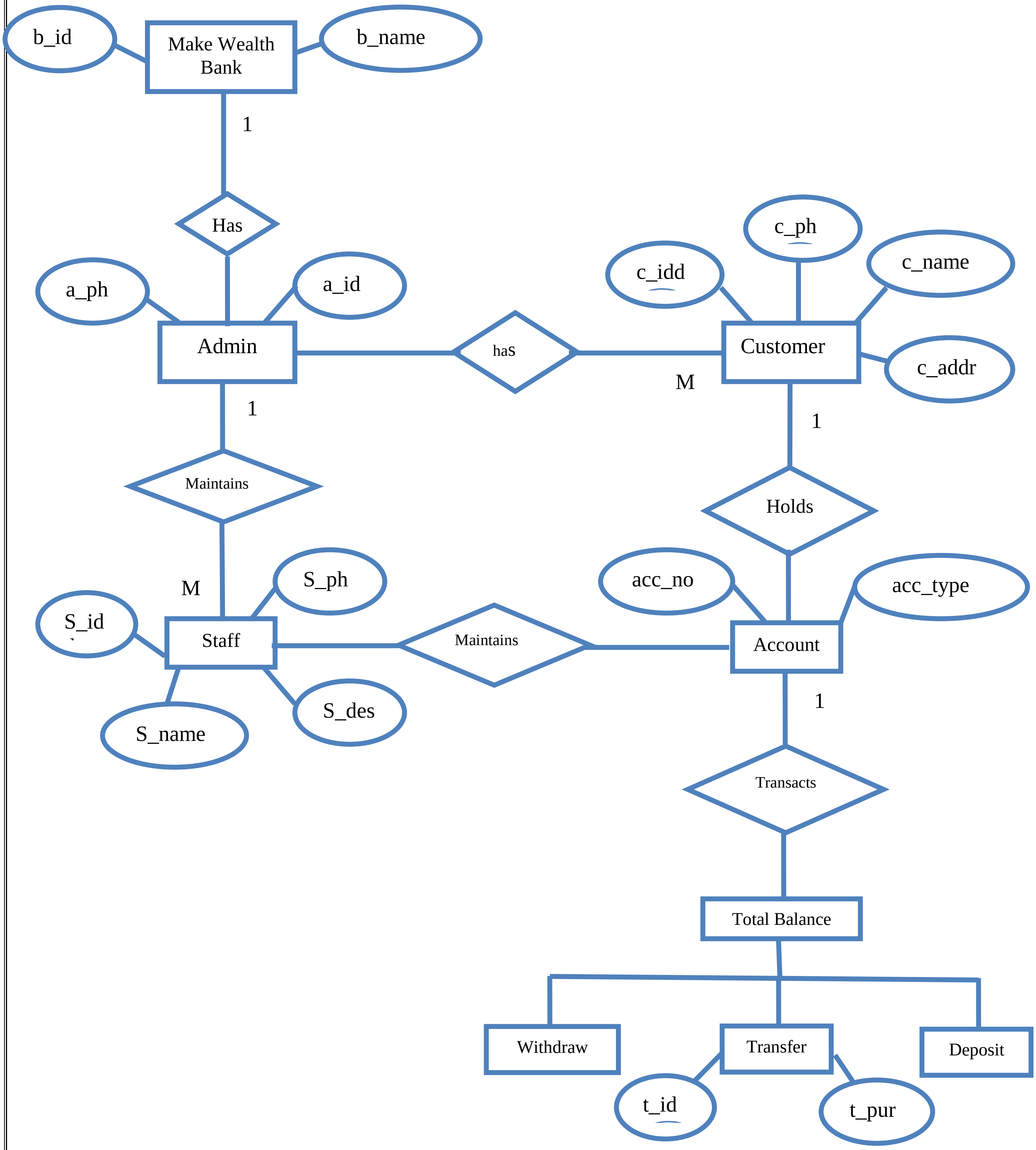
Hardware Requirement	
Process:	Intel core i3 1.80GHZ
RAM:	2GB or More
Monitor:	LCD monitor
Keyboard:	Normal keyboard
Mouse:	Compatible mouse

SYSTEM DESIGN

ERD Diagram :

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

E R- DIAGRAM

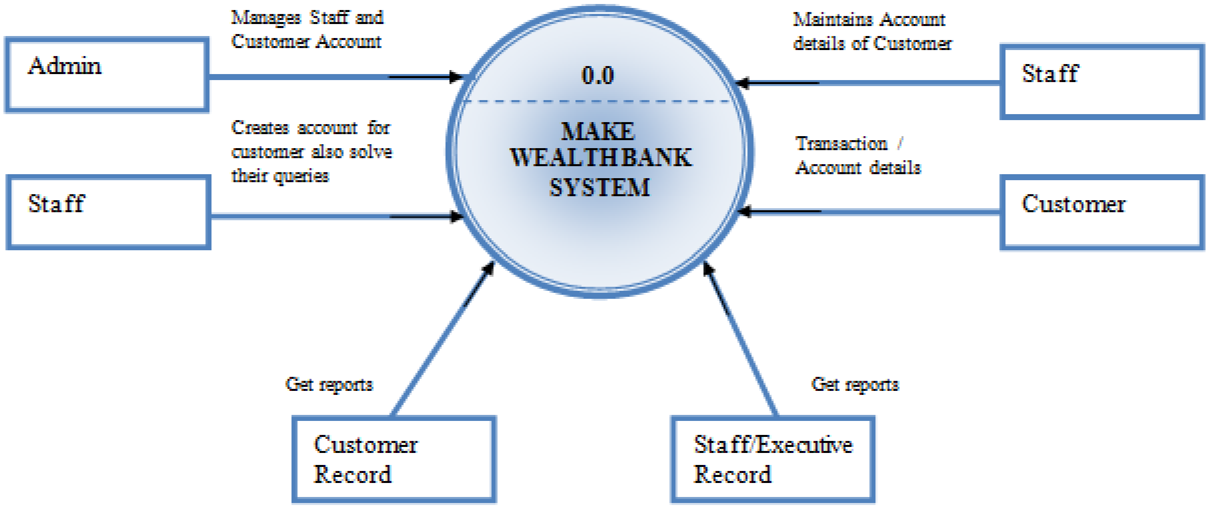

b_id
Make Wealth Bank
b_name
1
Has
a_ph
a_id
Admin
haS
c_idd
c_ph
c_name
Customer
c_addr
M
1
1
Maintains
Holds
M
S_ph
S_id
Staff
Maintains
acc_no
acc_type
Account
1
S_name
S_des
Transacts
Total Balance
Withdraw
Transfer
Deposit
t_id
t_pur

DFD

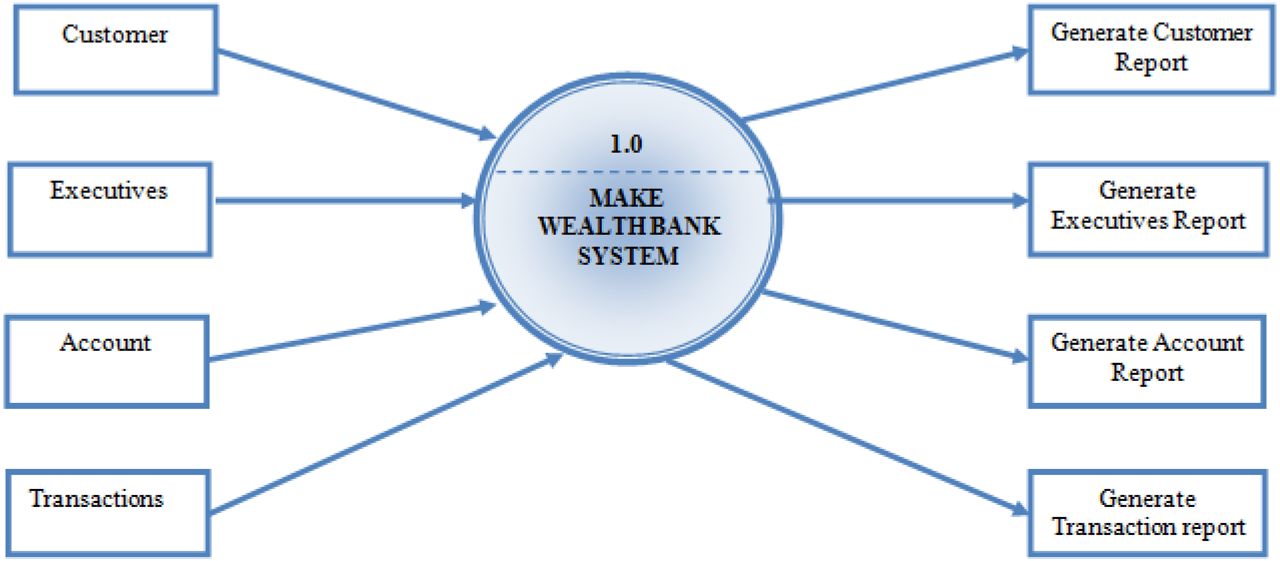
A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

DATA FLOW DIAGRAM

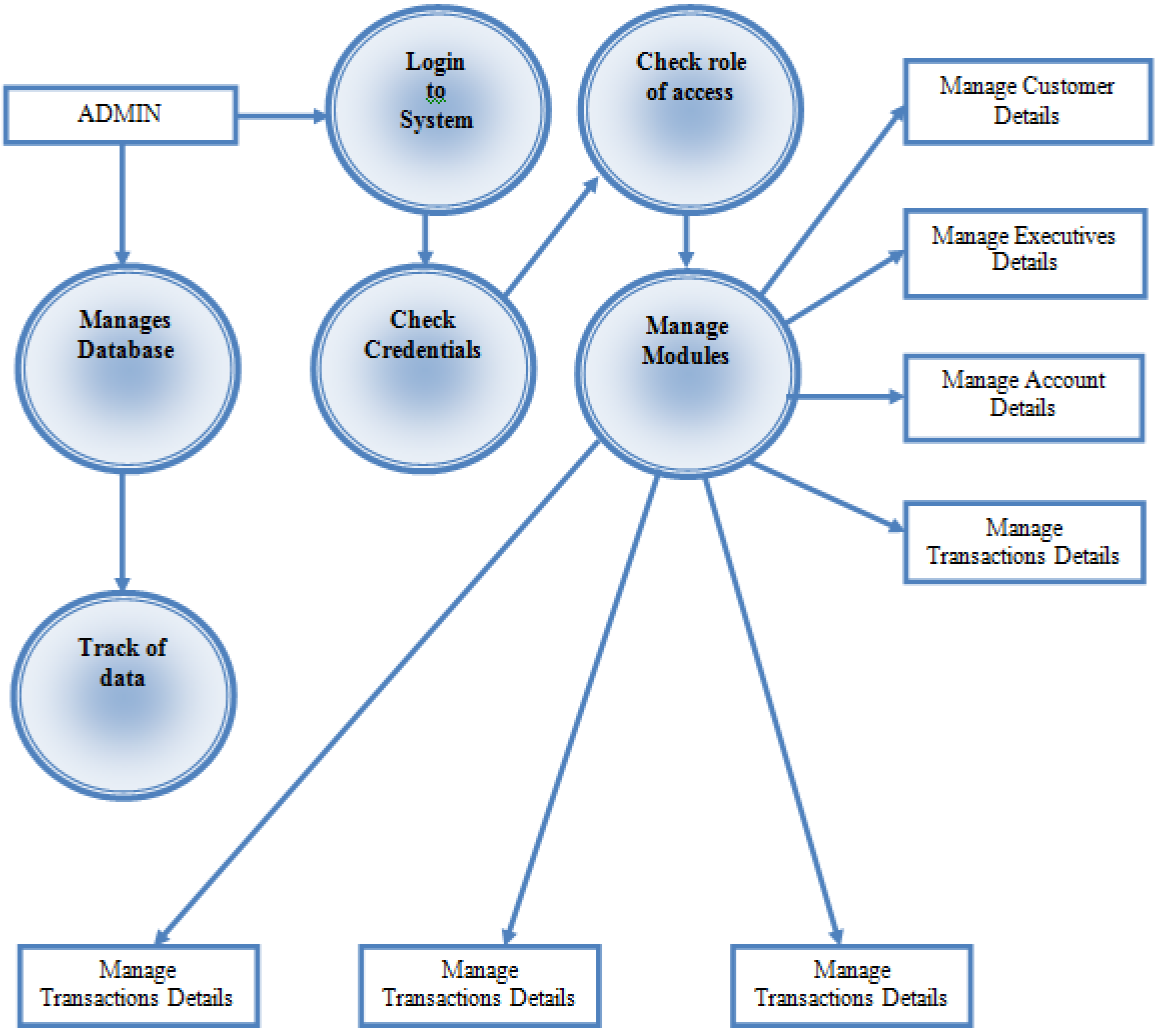
CONTEXT LEVEL DIAGRAM

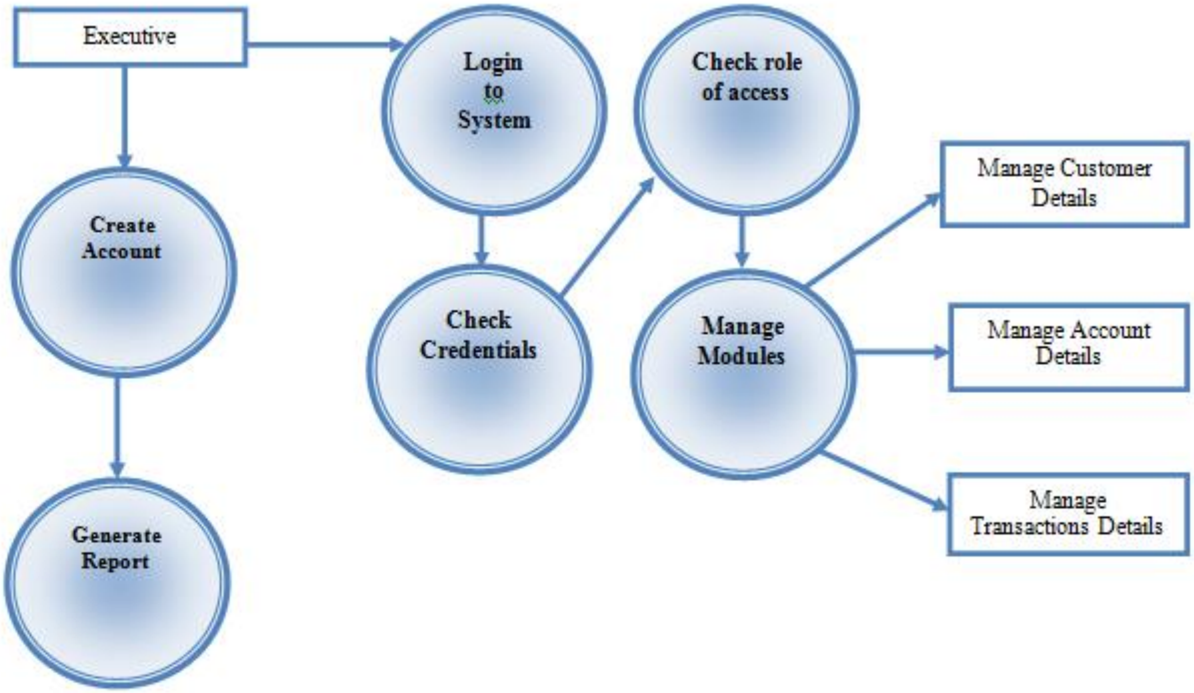

Admin
Manages Staff and Customer Account
Staff
Creates account for customer also solve their queries
0.0
MAKE WEALTH BANK SYSTEM
Maintains Account details of Customer
Staff
Transaction / Account details
Customer
Get reports
Customer Record
Get reports
Staff/Executive Record

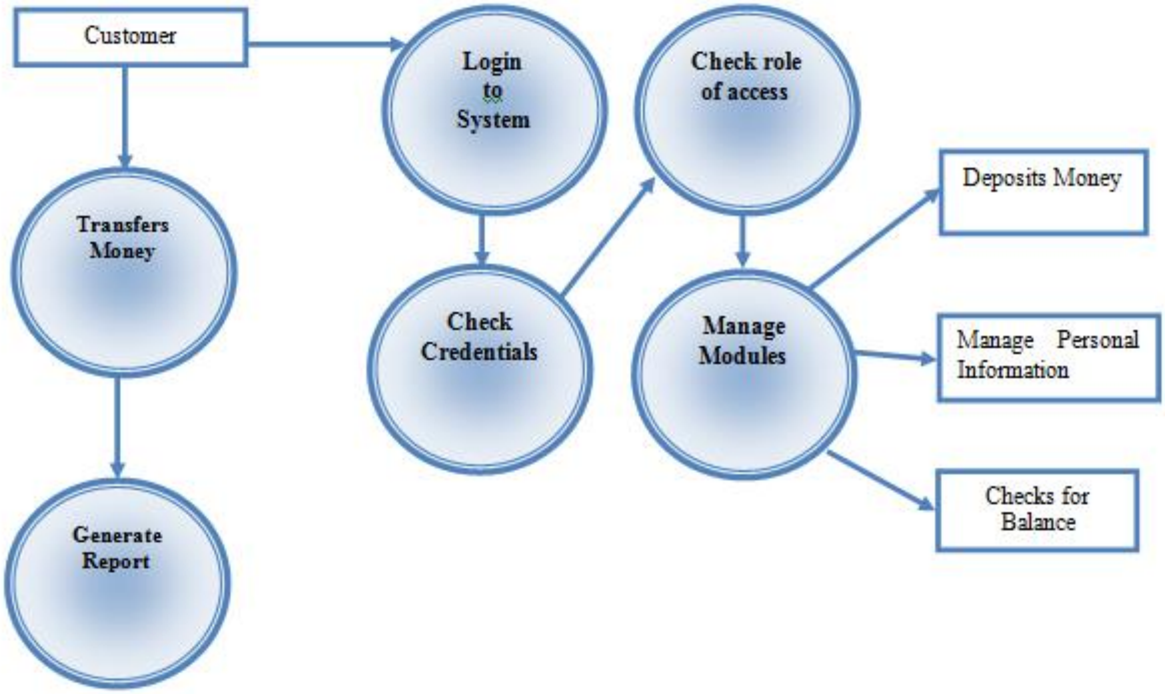
FIRST LEVEL DFD



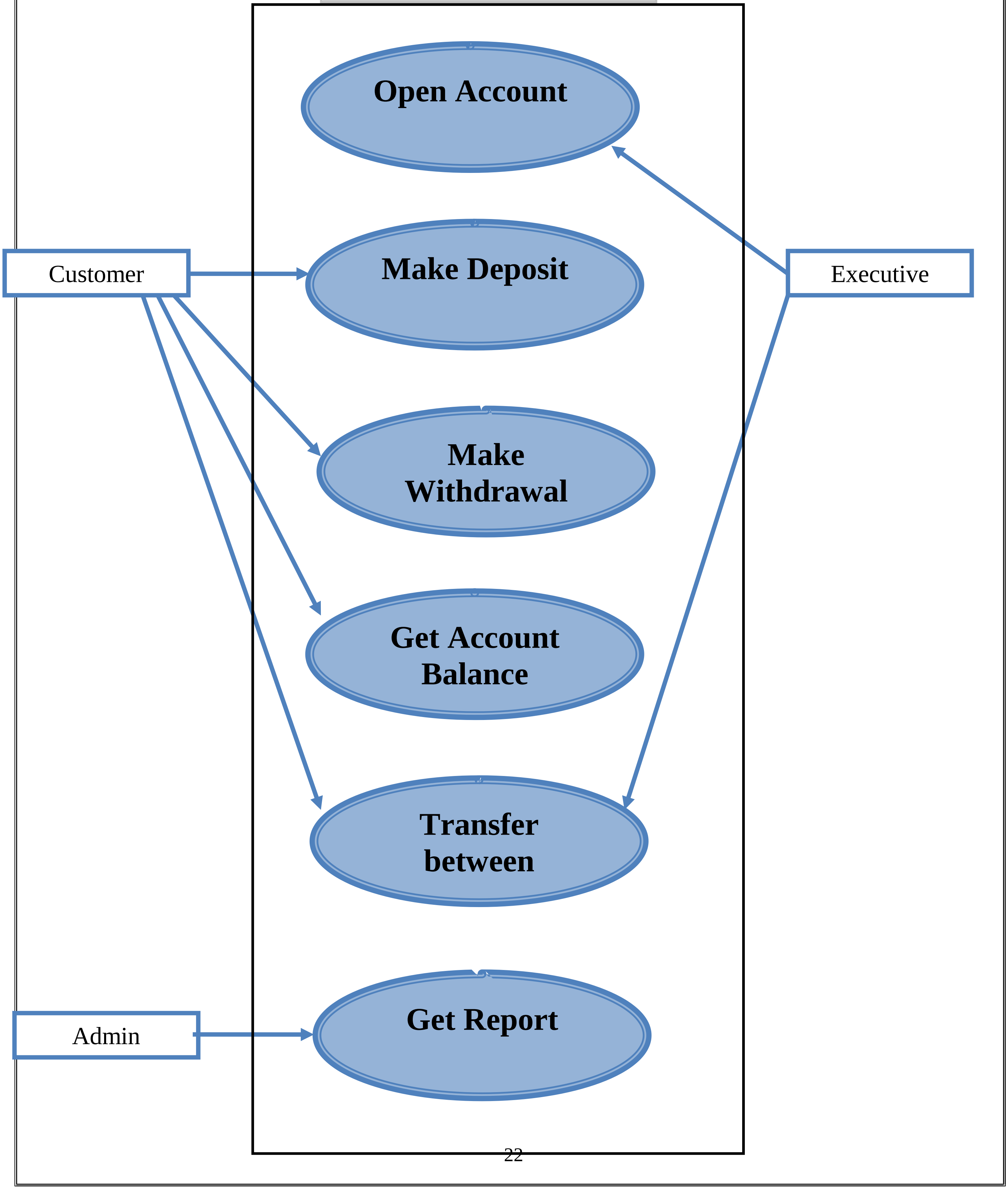
SECOND LEVEL DFD



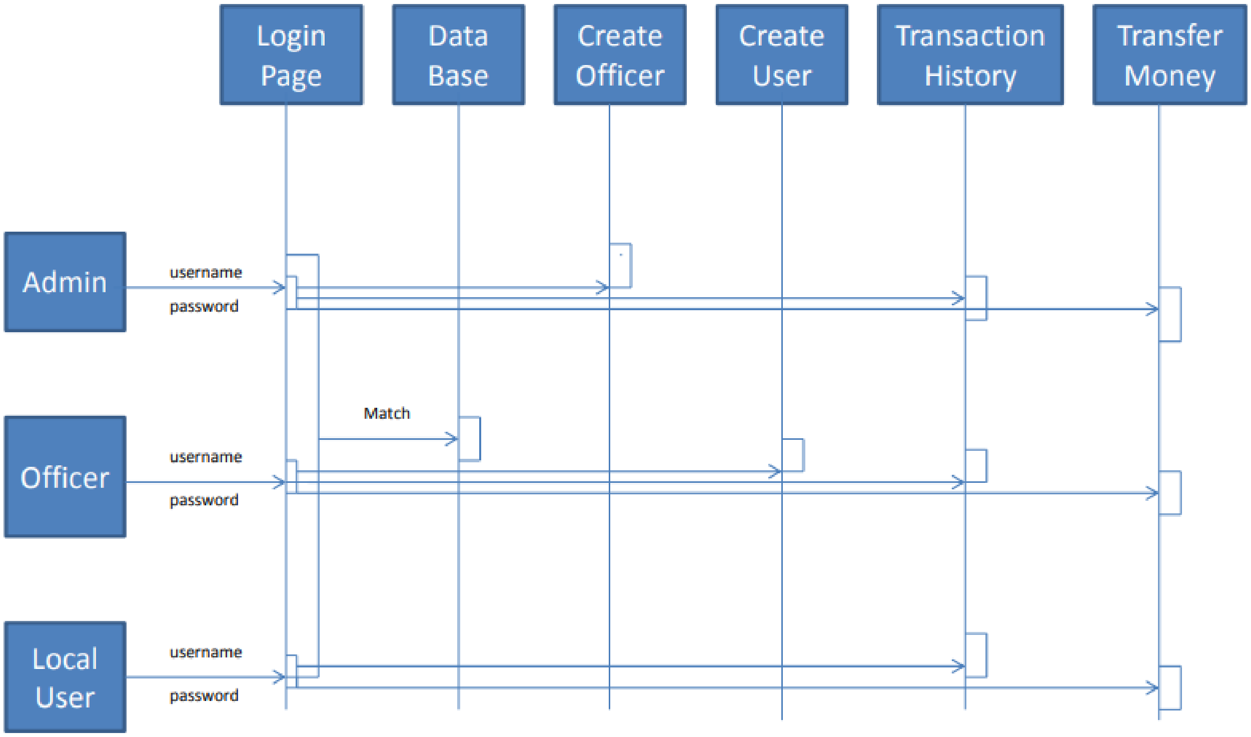

Executive
Login to System
Check role of access
Create Account
Check Credentials
Manage Modules
Manage Customer Details
Manage Account Details
Manage Transactions Details
Generate Report


Customer
Login to System
Check role of access
Transfers Money
Check Credentials
Manage Modules
Deposits Money
Manage Personal Information
Checks for Balance
Generate Report

USER CASE DIAGRAM



SEQUENCE DIAGRAM



A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata.

Data Dictionary

Sr.No	Field Name	Data Type	Width	Constrains	Table Name	Data description
1.	b_id	Number	10	Primary key	-	Bank ID
2.	b_name	Text	20	Not null	-	Bank name
3.	b_addr	Text	30	Not null	-	Bank r address
4.	b_phone	Number	10	Not null	-	Bank Phone number
5.	a_id	Number	30	Primary key	Admin	Admin ID
6.	a_ph	Number	10	Not null	Admin	Admin Phone Number
7.	c_id	Number	10	Primary	Customer	Customer ID
8.	c_name	Text	20	Not null	Customer	Customer Name
9.	c_addr	Text	30	Not null	Customer	Customer address
10.	c_phone	Number	10	Not null	Customer	Customer Phone Number
11.	c_acc	Number	20	Foreign key	Customer	Customer Account number
12.	s_id	Number	10	Primary key	Executive / Staff	Staff ID
13.	S_name	Text	10	Not null	Executive / Staff	Staff name
14.	s_addr	Text	20	Not null	Executive / Staff	Staff address
15.	s_ph	Number	10	Not null	Executive / Staff	Staff Phone number
16.	t_id	Number	10	Primary key	Transaction	Transaction ID
17.	t_date	Date	10	Not null	Transaction	Transaction Date
18.	t_amt	Currency	10	Not null	Transaction	Transaction Amount
19.	t_pur	Text	10	Not null	Transaction	Transaction purpose

NORMALIZED DATA

Database normalization is the process of structuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. It was first proposed by Edgar F. Codd as part of his relational model.

TABLE: ADMIN

SR.NO	Field Name	Data Type	Width	Constraints
1	Username_id	Number	10	Primary Key
2	Password	Number	8	Not null

TABLE: EXECUTIVE / STAFF

SR.NO	Field Name	Data Type	Width	Constraints
1	s_id	Number	10	Primary key
2	s_name	Text	20	Not null
3	s_addr	Text	30	Not null
6	s_phone	Number	10	Not null

TABLE: Customer				
SR.NO	Field Name	Data Type	Width	Constraints
1	c_id	Number	10	Primary key
2	c_name	Text	20	Not null
3	c_addr	Text	30	Not null
6	C_account	Number	10	Foreign key

TABLE: Transaction				
SR.NO	Field Name	Data Type	Width	Constraints
1	t_id	Number	10	Primary key
2	t_date	Text	20	Not null
3	t_pur	Text	30	Not null
4	c_id	Number	10	Foreign Key

Test Procedures and Implementation

Test procedure

The software testing is the critical element of software quality assurance and represents the ultimate review of the software design and coding. The main objective of the testing is to find an error and to uncover the errors that are not yet discovered.

The increasing visibility of software as a system element and the attendant cost associated with a software failure and motivating forces for well planned, through testing. It is not unusual for a software development organization to expand between 30% to 40% of project effort on testing. In the extreme, testing of human related software can cost 3-5 times as much as all other software engineering activities combined. The testing phase involves the testing of the system using various test data, preparation of the test data plays a vital role in the system testing after preparing the test data, errors were found and corrected by using the following the testing steps and correction are recorded for future reference. Thus, a series of testing is performed on the system before it is ready for implementation.

After completion of system analysis, design, and coding through testing of the system was carried out in a systematic approach, the main objectives of the system are

- To ensure that the operations of the system will perform as per the specification.
- To make sure that the system meets the user requirement during the operations.
- To cross check when correct input are filled into the system output are correct.
- To make sure that during the operation incorrect inputs and the outputs will be detected.

In testing process the number of strategies have been used as mentioned below,

- Unit Testing
- Integration Testing
- Validation Testing
- Black Box Testing
- User acceptance Testing

Unit Testing

Unit testing focuses verification efforts on the smallest unit of the software design. Using the system test plan, prepared in the design phase of the system development as a guide, important control paths are tested to uncover errors within the boundary of the module. The interface of each of the modules was tested to ensure proper flow of information into and out of the module under consideration. Each module will be tested individually so as to make the individual component error free. Also other attached modules will be error free.

Integration Testing:

Each module will be tested on its effect on other modules by integrating the modules. This will remove further errors from the system and may also result in some changes in the individual module.

Validation Testing

At the culmination of the integration testing the software was completely assembled as a package, interfaces were uncovered, and a final series of software validation testing began. Here we test the system function manner that can be reasonable by the customer, the system was tested against system requirement specification.

Black Box Testing:

After performing validation testing, the next phase is the output test of the system, since no system code is useful if it does not produce the desired output in desired format. By considering the format of the report/output, report/output is generated or displayed and tested.

User Acceptance Testing:

User acceptance testing is used to determine whether the software is fit for the user to use. The System under consideration was listed for user acceptance by keeping constant touch with the prospective user of the system at the time of design, development and making changes whenever required.

Test Case:

- Title: Test case for Login.
- Objective: To check that user properly logged in.

Test Case Id	Test Type	Test Case Name	Steps to be Followed	Expected Result	Actual Result	Status	Priority
1	The application should be installed properly & Accessible.	Test case for Login	1. Enter Username / user id 2. Enter password 3. Click Login Button	It Shows Dashboard	It Shows Dashboard	Pass	2
2			1. Enter username / user id 2. Click Login Button	Check username and password	Check username and password	Pass	

3.			1. Enter password 2. click Login Button	Check username and password	Check username and password	Pass	

Title: Test case for user Registration.

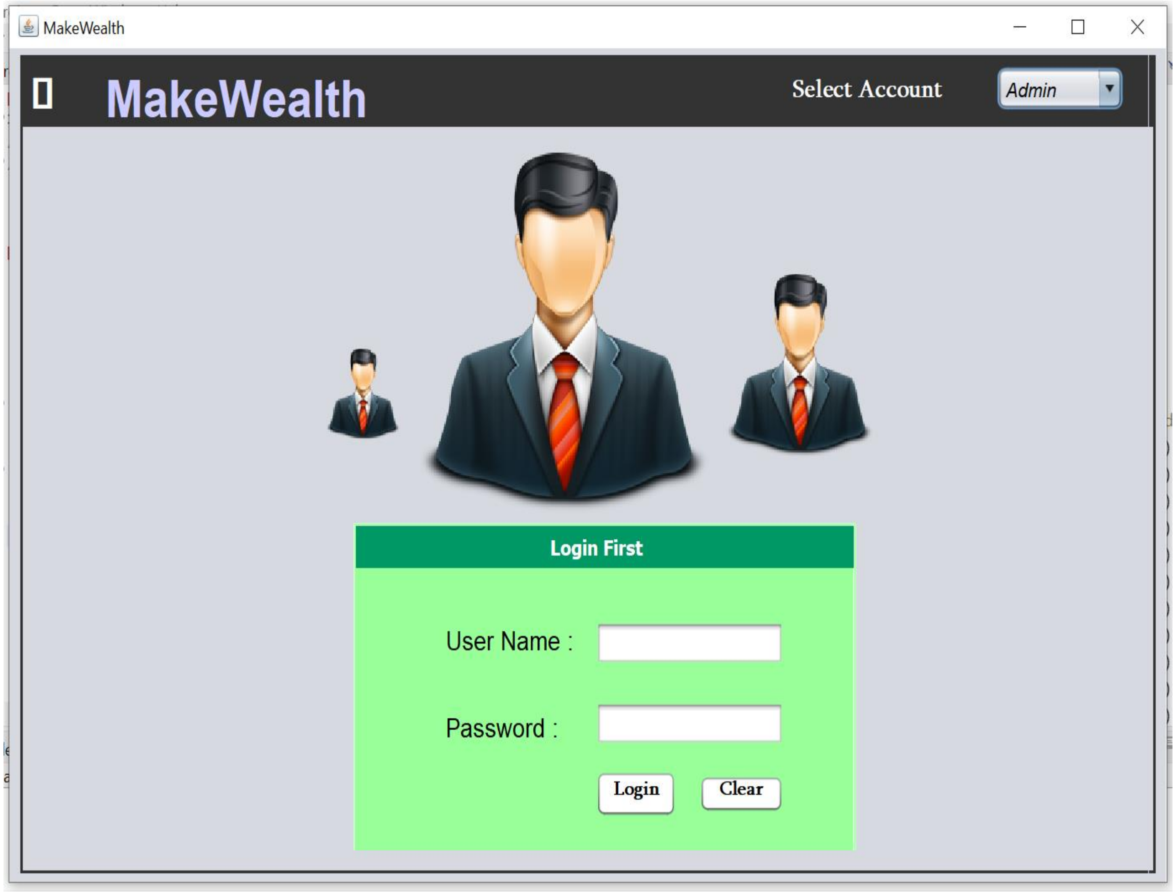
Objective: To check how new users are properly added.

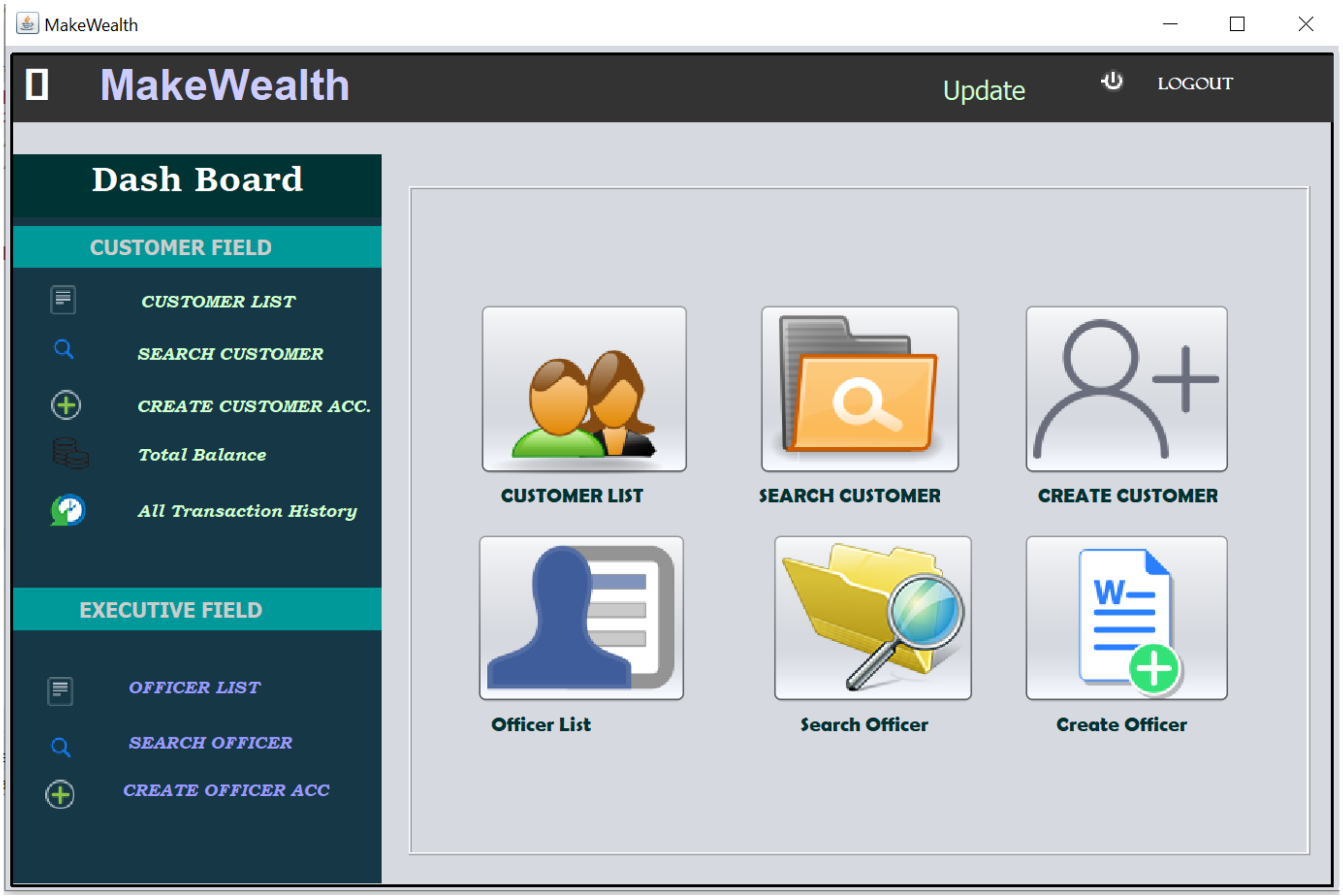
Test Case Id	Test Type	Test Case Name	Steps to be Followed	Expected Result	Actual Result	Status	Priority

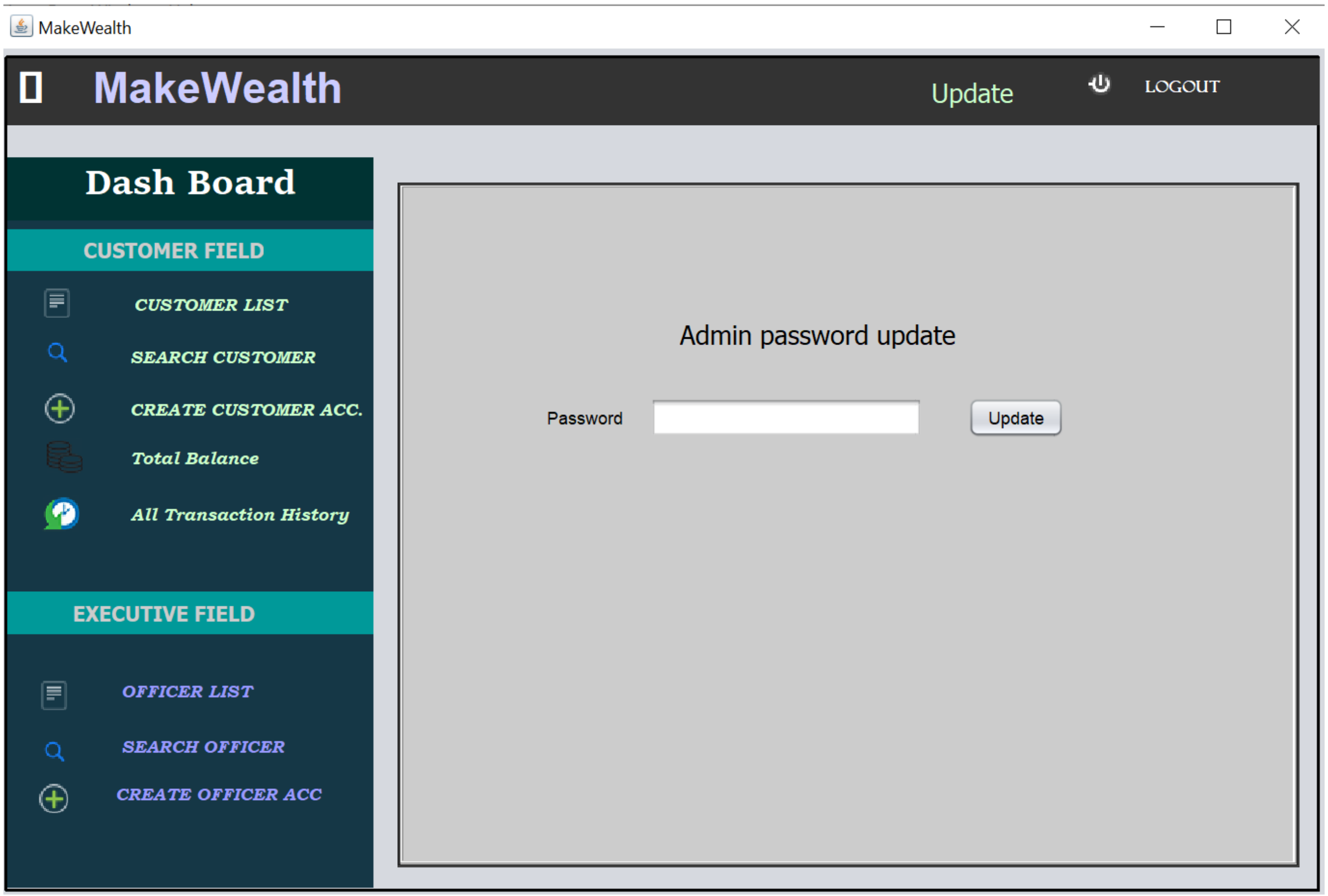
1.	The application should be installed properly & Accessible.	Test case for user Registration. (New account holder)	1.Enter User Name 2.Enter password 3.Enter join date 4.Enter account type 5.Enter nationality 6.Enter Name 7. Enter phone number 8.Enter Date of birth 9. Enter address Register button	It shows all fields are compulsory Message.	It shows all fields are compulsory Message.	Pass	2
2.			1.Enter Name 2.click Register button	It shows invalid name. accept only Characters.	It shows invalid name. accept only Characters.	Pass	

3.			1. Enter account id 2.click Register button	It shows id already exists (if already in DB)	It shows id already exists (if already in DB)	pass	
4.			1.Enter Join date or birth date 2.click Register button	It shows invalid date format. Should be (yyyy/mm/day)	It shows invalid date format. Should be (yyyy/mm/day)	Pass	

CODE DESIGN

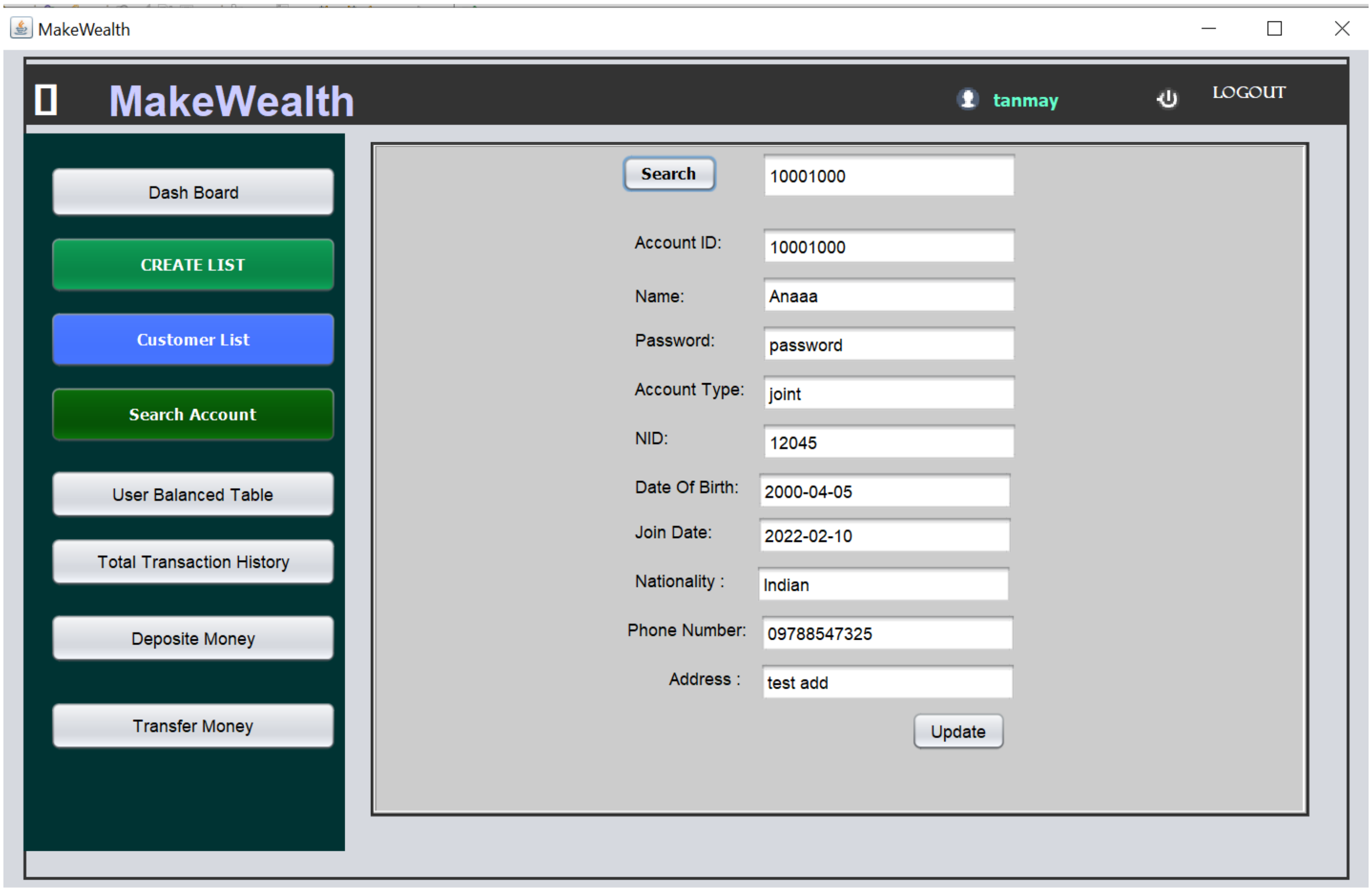

MakeWealth
MakeWealth
Select Account
Admin
Login First
User Name :
Password :
Login
Clear


MakeWealth
MakeWealth
Update
LOGOUT
Dash Board
CUSTOMER FIELD
CUSTOMER LIST
SEARCH CUSTOMER
CREATE CUSTOMER ACC.
Total Balance
All Transaction History
EXECUTIVE FIELD
OFFICER LIST
SEARCH OFFICER
CREATE OFFICER ACC
CUSTOMER LIST
SEARCH CUSTOMER
CREATE CUSTOMER
Officer List
Search Officer
Create Officer



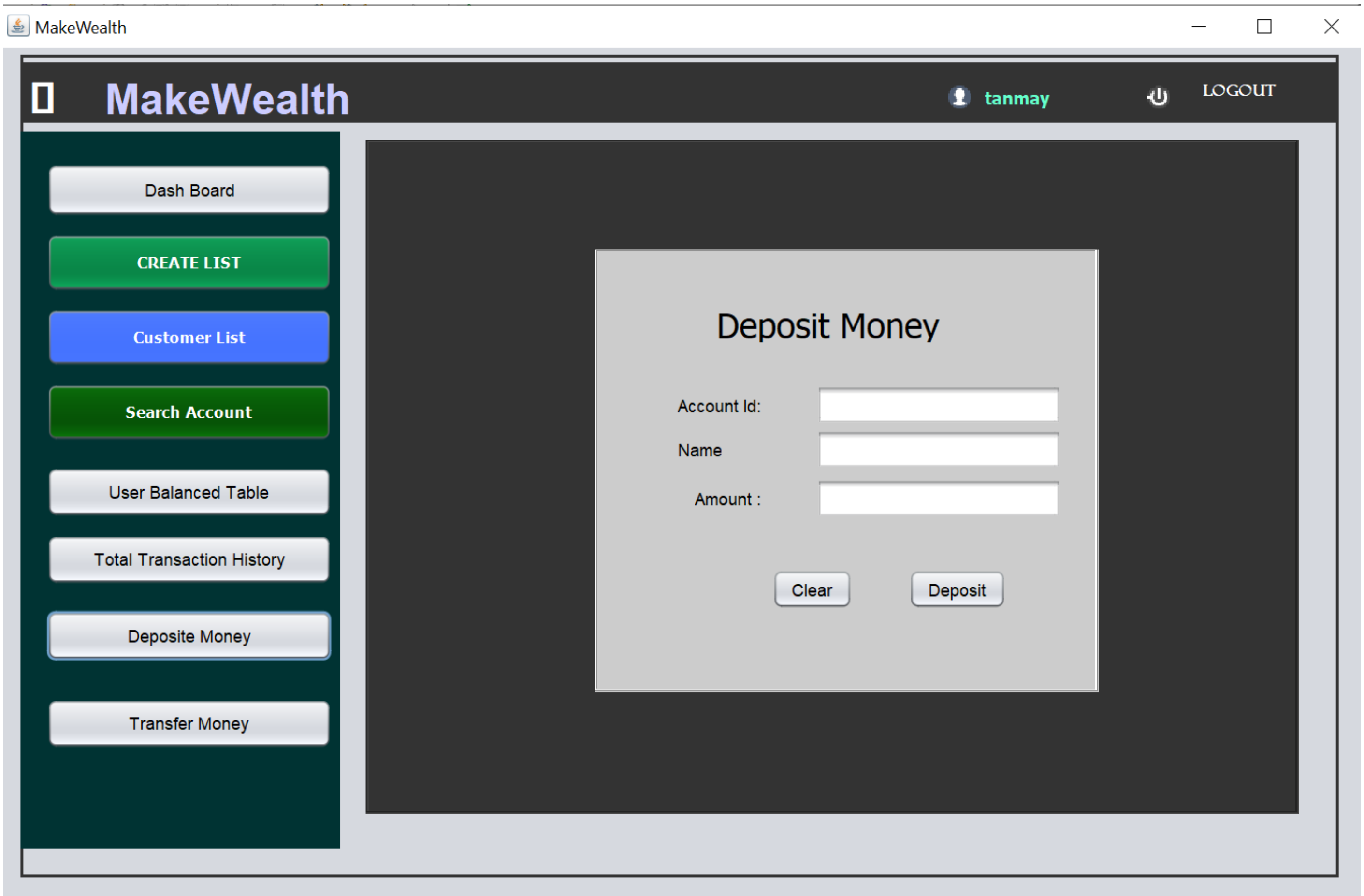
Admin password update

Password



Search: 10001000

Account ID:	10001000
Name:	Anaaa
Password:	password
Account Type:	joint
NID:	12045
Date Of Birth:	2000-04-05
Join Date:	2022-02-10
Nationality :	Indian
Phone Number:	09788547325
Address :	test add

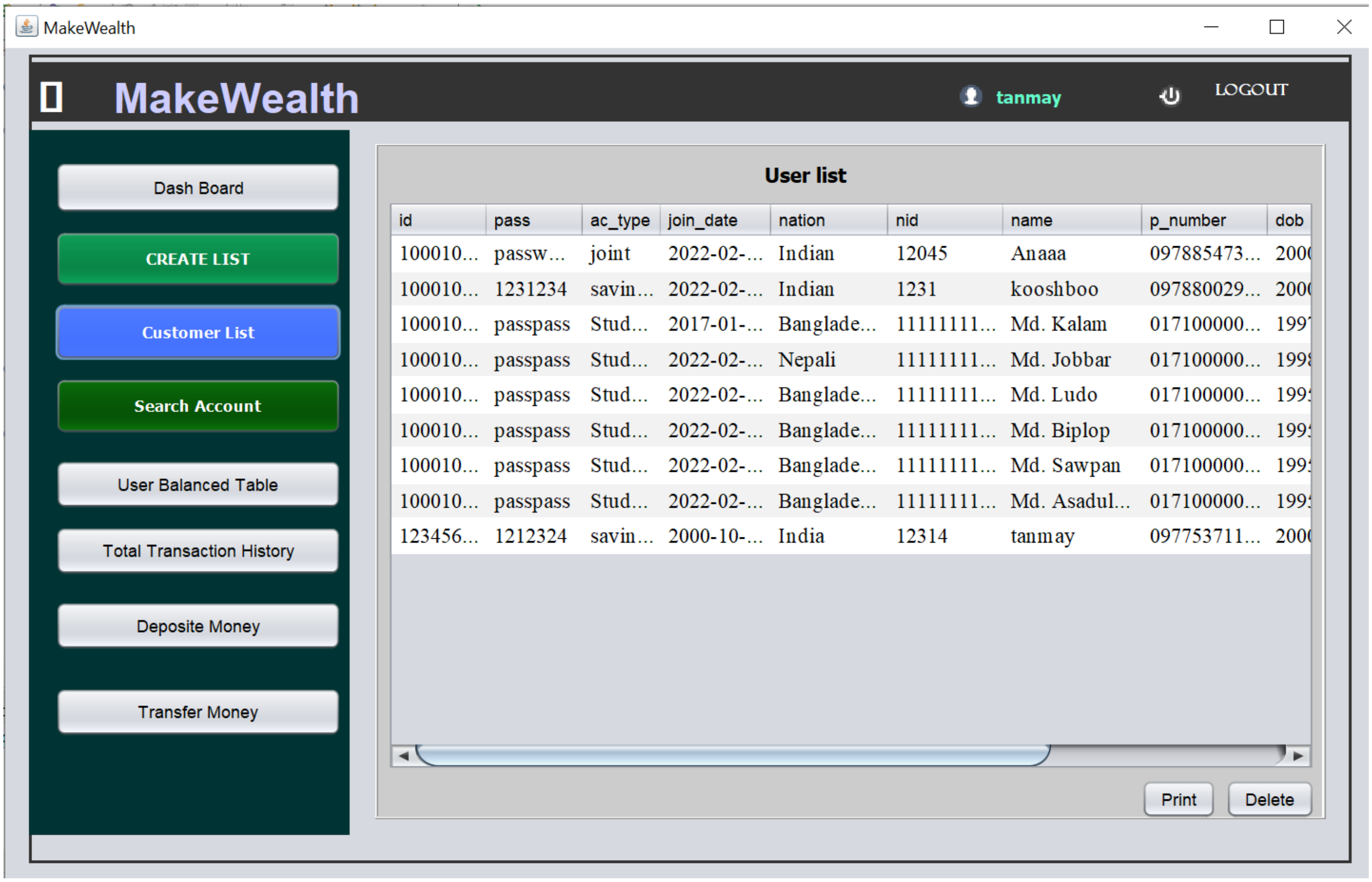


Deposit Money

Account Id:

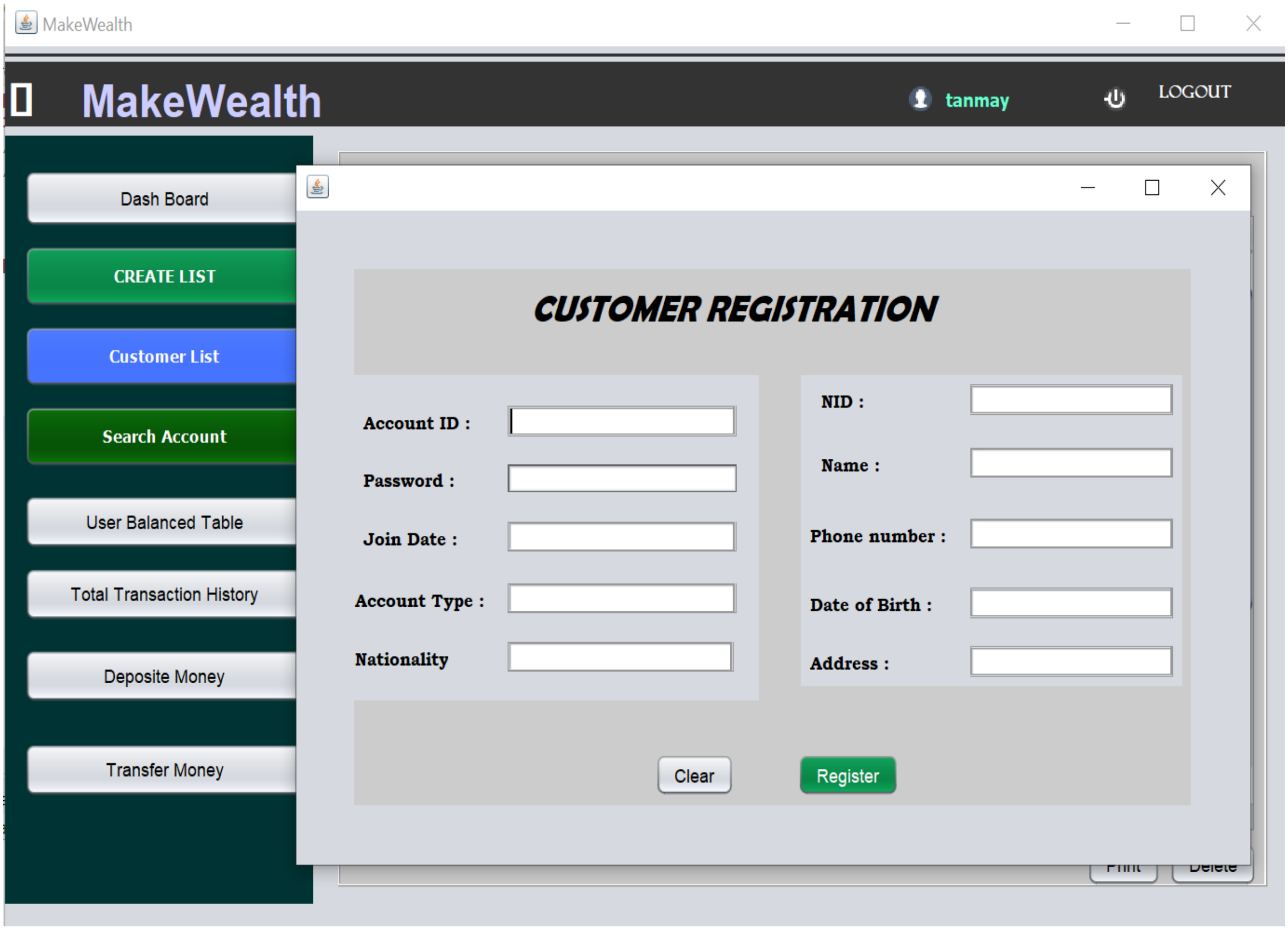
Name

Amount :



User list

id	pass	ac_type	join_date	nation	nid	name	p_number	dob
100010...	passw...	joint	2022-02-...	Indian	12045	Anaaa	097885473...	2000
100010...	1231234	savin...	2022-02-...	Indian	1231	kooshboo	097880029...	2000
100010...	passpass	Stud...	2017-01-...	Banglade...	11111111...	Md. Kalam	017100000...	1997
100010...	passpass	Stud...	2022-02-...	Nepali	11111111...	Md. Jobbar	017100000...	1998
100010...	passpass	Stud...	2022-02-...	Banglade...	11111111...	Md. Ludo	017100000...	1995
100010...	passpass	Stud...	2022-02-...	Banglade...	11111111...	Md. Biplop	017100000...	1995
100010...	passpass	Stud...	2022-02-...	Banglade...	11111111...	Md. Sawpan	017100000...	1995
100010...	passpass	Stud...	2022-02-...	Banglade...	11111111...	Md. Asadul...	017100000...	1995
123456...	1212324	savin...	2000-10-...	India	12314	tanmay	097753711...	2000



CUSTOMER REGISTRATION

Account ID :

Password :

Join Date :

Account Type :

Nationality

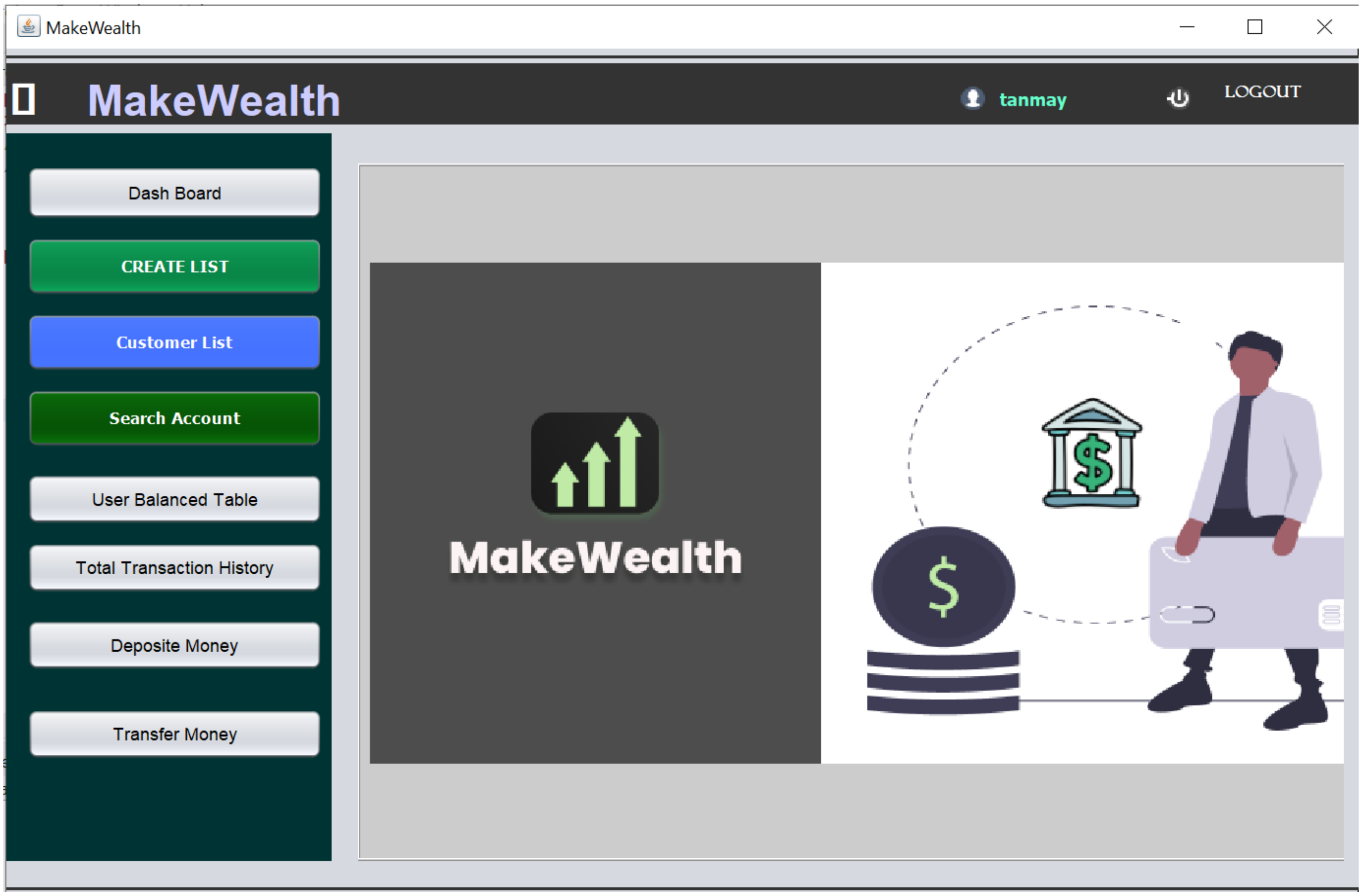
NID :

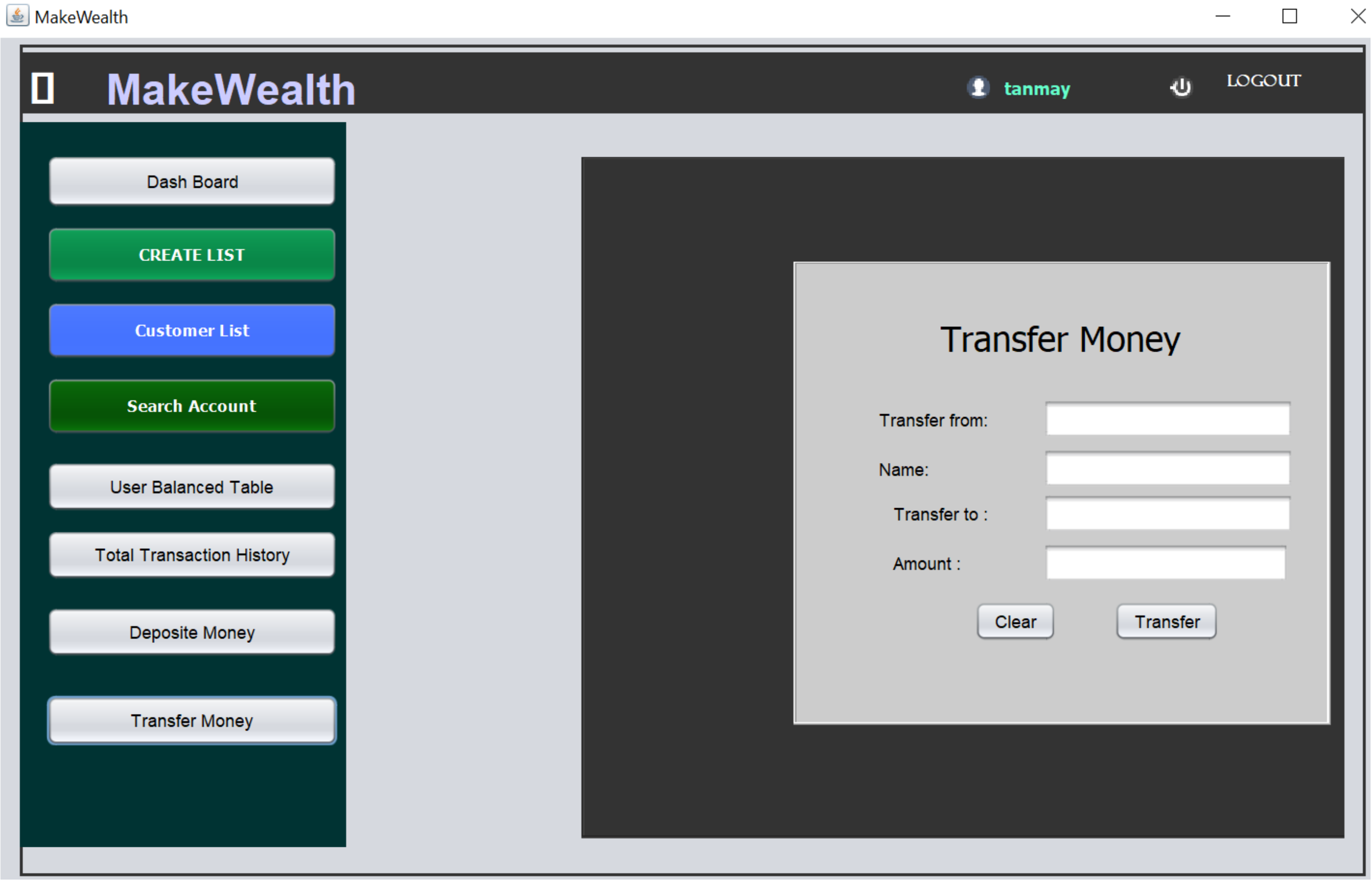
Name :

Phone number :

Date of Birth :

Address :


MakeWealth
MakeWealth
tanmay
LOGOUT
Dash Board
CREATE LIST
Customer List
Search Account
User Balanced Table
Total Transaction History
Deposite Money
Transfer Money
MakeWealth



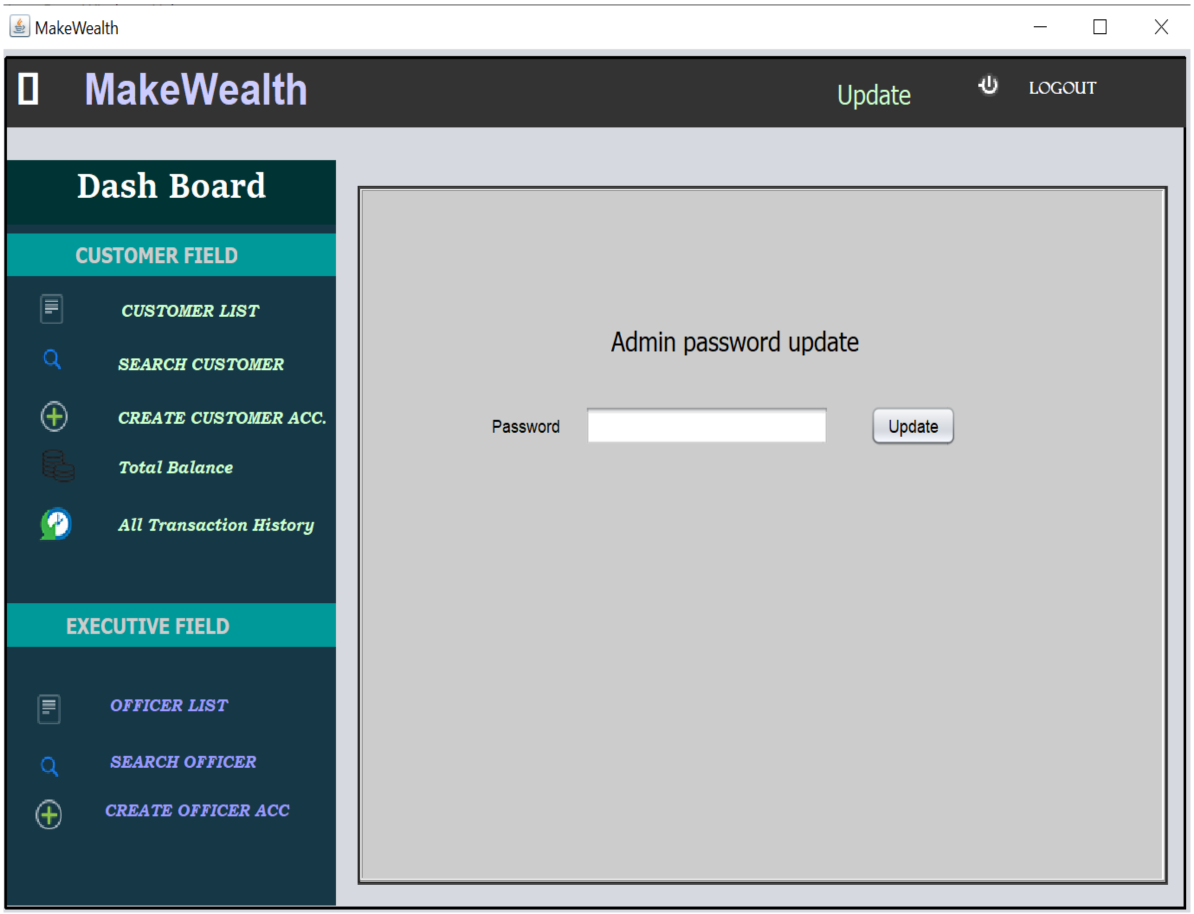
Transfer Money

Transfer from:

Name:

Transfer to :

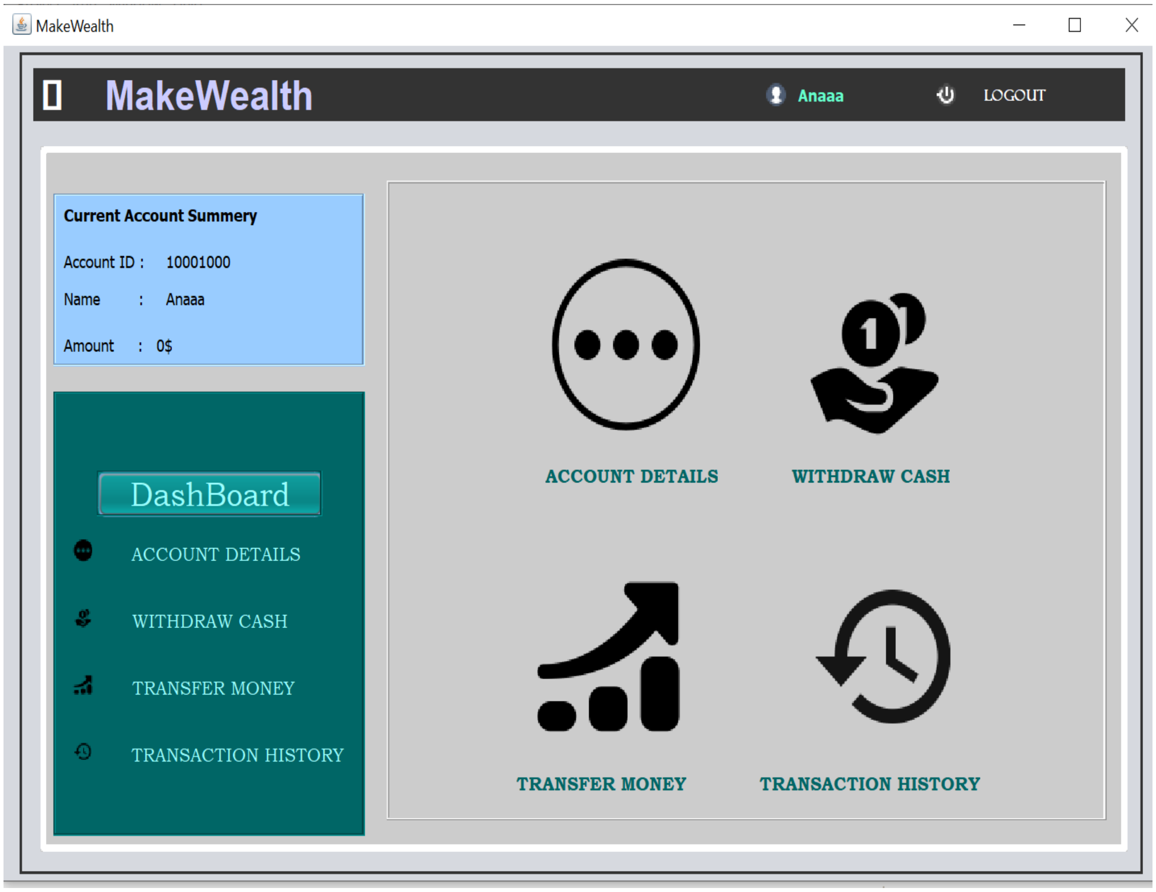
Amount :



Admin password update

Password

\

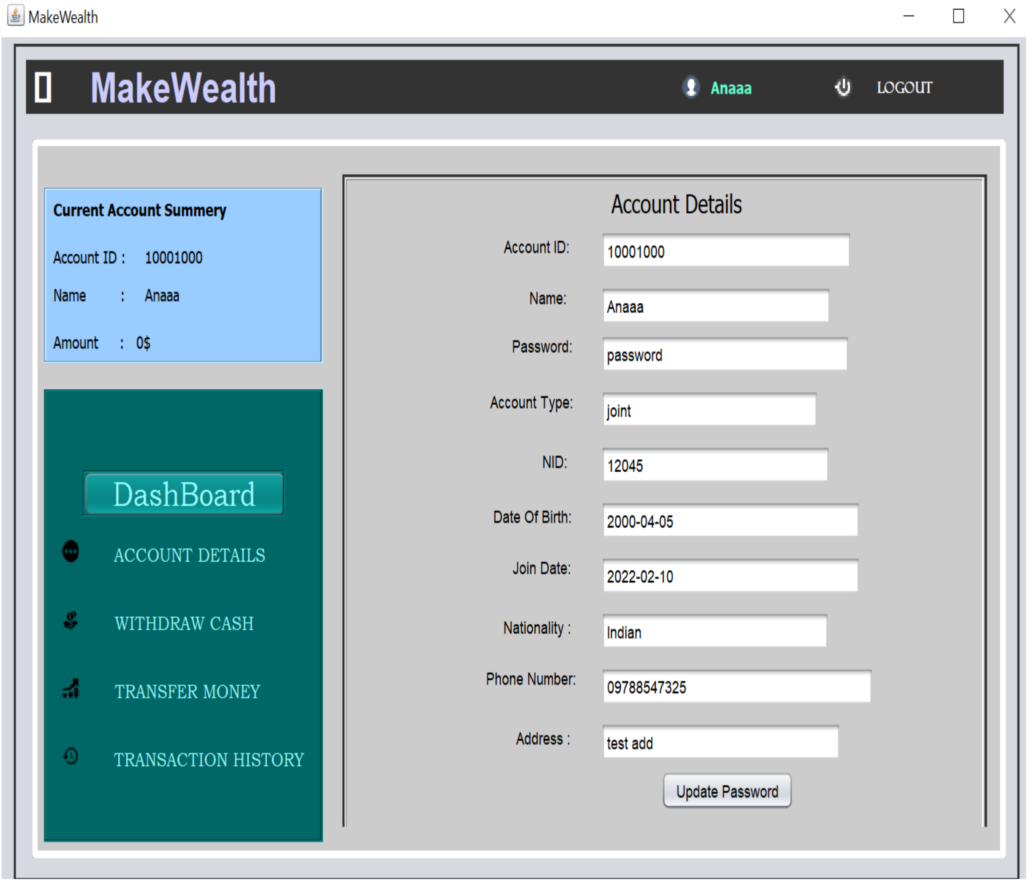


Current Account Summery

Account ID : 10001000

Name : Anaaa

Amount : 0$



Current Account Summery

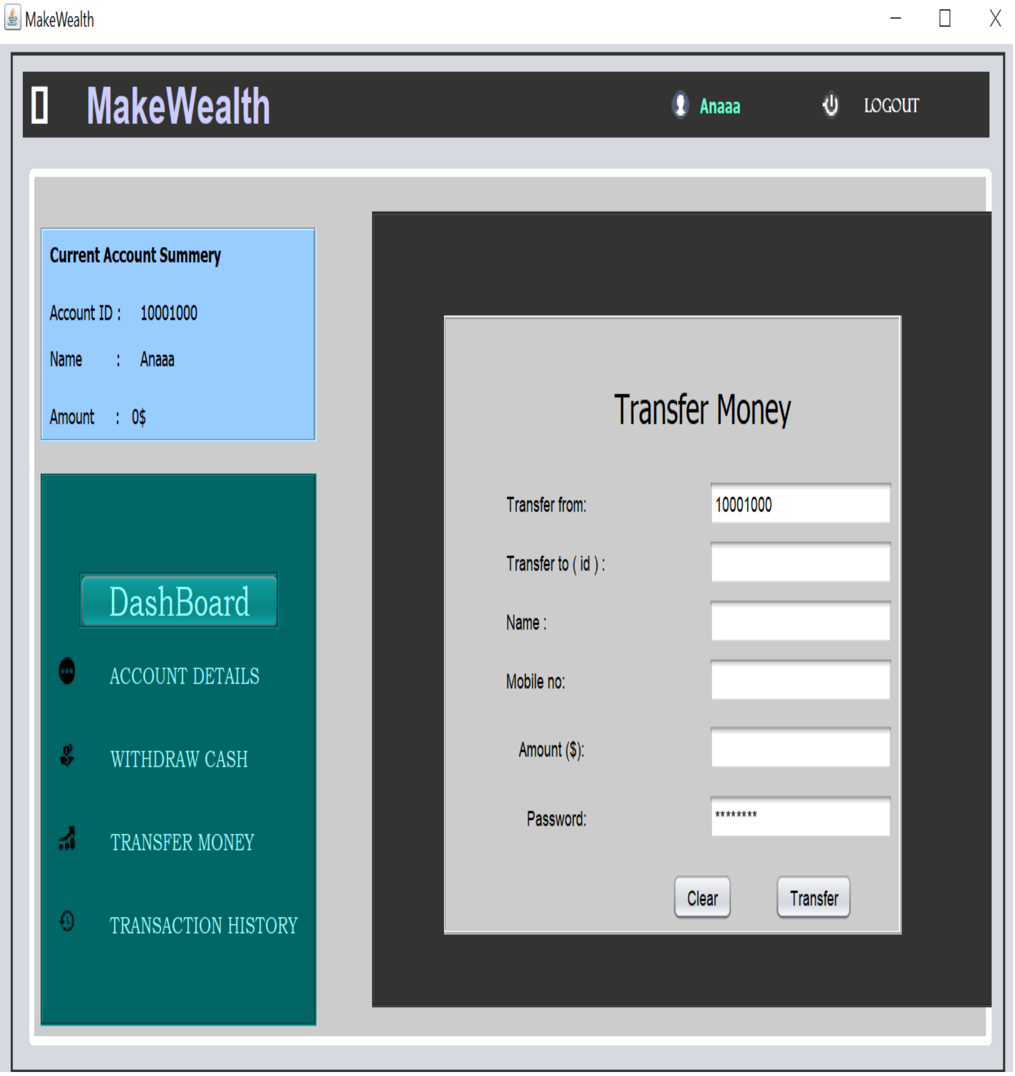
Account ID : 10001000

Name : Anaaa

Amount : 0$

Account Details

Account ID:	10001000
Name:	Anaaa
Password:	password
Account Type:	joint
NID:	12045
Date Of Birth:	2000-04-05
Join Date:	2022-02-10
Nationality :	Indian
Phone Number:	09788547325
Address :	test add



Current Account Summery

Account ID : 10001000

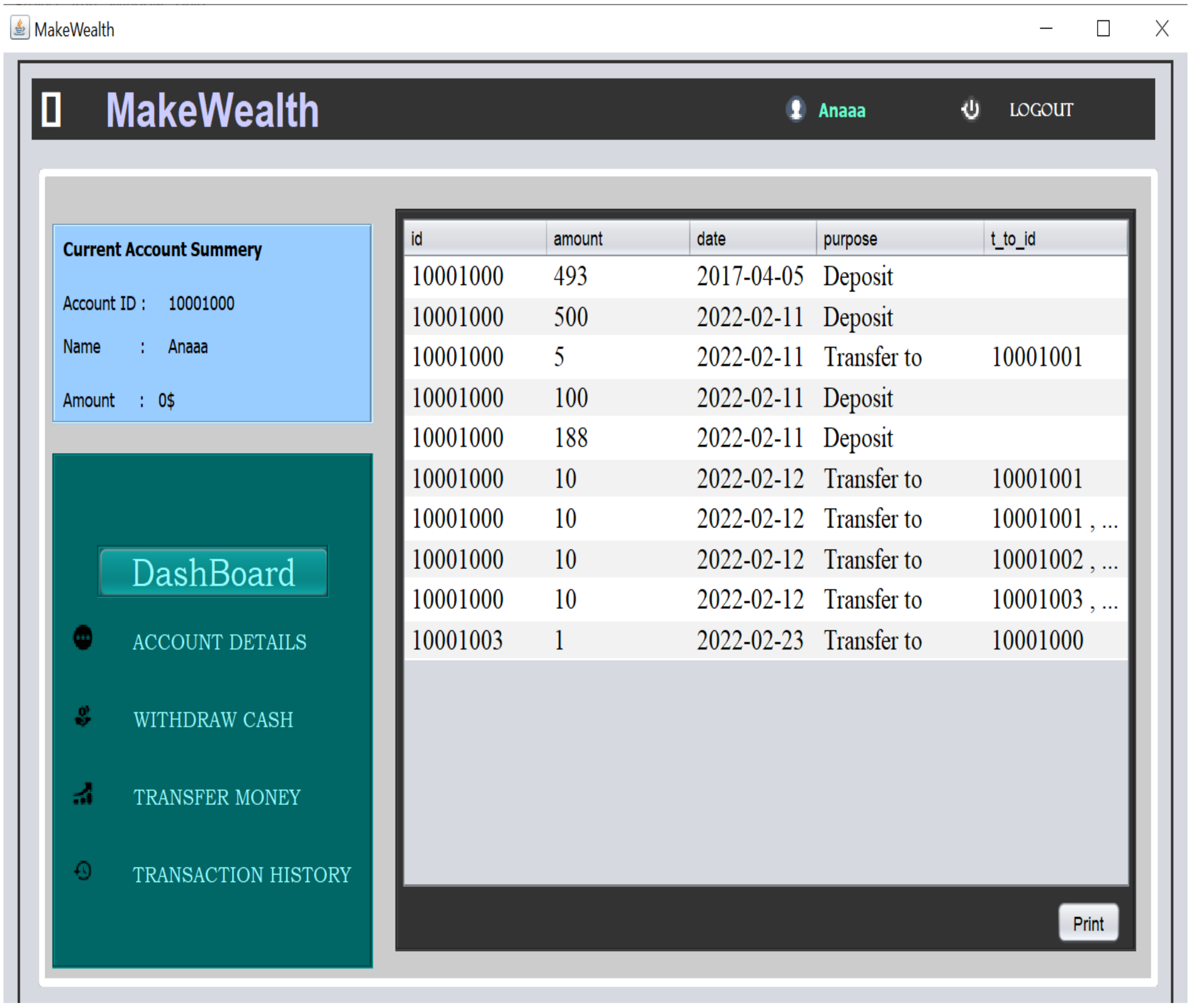
Name : Anaaa

Amount : 0$

Transfer Money

Transfer from:	10001000
Transfer to (id) :	
Name :	
Mobile no:	
Amount ($):	
Password:	********

TEST REPORTS



Current Account Summery

Account ID : 10001000

Name : Anaaa

Amount : 0$

id	amount	date	purpose	t_to_id
10001000	493	2017-04-05	Deposit	
10001000	500	2022-02-11	Deposit	
10001000	5	2022-02-11	Transfer to	10001001
10001000	100	2022-02-11	Deposit	
10001000	188	2022-02-11	Deposit	
10001000	10	2022-02-12	Transfer to	10001001
10001000	10	2022-02-12	Transfer to	10001001 , ...
10001000	10	2022-02-12	Transfer to	10001002 , ...
10001000	10	2022-02-12	Transfer to	10001003 , ...
10001003	1	2022-02-23	Transfer to	10001000

USER REQUIREMENTS

Functional Requirements

● In software engineering, a functional requirement defines a function of a software system or its component.

● A function is described as a set of inputs, the behavior, and outputs

● Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases

● Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

● Generally, functional requirements are expressed in the form "system must do <requirement>", while non-functional requirements are "system shall be <requirement>". The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture. ● As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

Non Functional Requirements

➢ **Product Requirements**

Usability requirements

Usability is the ease of use and learning ability of a human-made object. The object of use can be a software application, website, book, tool, machine, process, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. Usability includes methods of measuring usability, such as needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed. Usability differs from user satisfaction and user experience because usability also considers usefulness.

Reliability requirements

Reliability deals with the study, evaluation, and life-cycle management of reliability: the ability of a system or component to perform its required functions under stated conditions for a specified period of time. Reliability engineering is a sub-discipline within systems engineering. Reliability is theoretically defined as the probability of failure, the frequency of failures, or in terms of availability, a probability derived from reliability and maintainability. Maintainability and maintenance may be defined as a part of reliability engineering. Reliability plays a key role in cost-effectiveness of systems.

Portability requirement

Portability in high-level computer programming is the usability of the same software in different environments. The pre-requirement for portability is the generalized abstraction between the application logic and system interfaces. When software with the same functionality is produced for several computing platforms, portability is the key issue for development cost reduction.

- Transferring installed program files to another computer of basically the same architecture. - Reinstalling a program from distribution files on another computer of basically the same architecture. Efficiency requirements

Resource consumption for a given load describes efficiency of product and web site or a working of software .

Performance requirements

Performance metrics include availability, response time, channel capacity, latency, completion time, service time, bandwidth, throughput, relative efficiency, scalability, performance per watt, compression ratio, instruction path length and speed up.

Short response time for a given piece of work

- High throughput (rate of processing work)
- Low utilization of computing resource(s)
- High availability of the computing system or application
- Fast (or highly compact) data compression and decompression
- High bandwidth / short data transmission time

➢Organizational Requirements

Delivery requirements

Delivery requirements include details of delivery of product on time and as per client requirements. The product should be delivered on prescribed standards.

Implementation requirements

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

Standard requirements

The project should be developed as per standard format specified by IEEE. Typical platforms include a computer architecture, operating system, programming languages and related user interface. The product should be developed as per client's standard requirements.

➢ **External Requirements**

Interoperability requirements

Interoperability is a property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation.

The IEEE Glossary defines interoperability as:

the ability of two or more systems or components to exchange information and to use the information that has been exchanged

Legislative requirements

In the proprietary software industry, an end-user license agreement or software license agreement is the contract between the licensor and purchaser, establishing the purchaser's right to use the software. The license may define ways under which the copy can be used.Software companies

often make special agreements with large businesses and government entities that include support contracts and specially drafted warranties.

Privacy requirements

The term "privacy" means many things in different contexts. Different people, cultures, and nations have a wide variety of expectations about how much privacy a person is entitled to or what constitutes an invasion of privacy.

Privacy is the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively. The boundaries and content of what is considered private differ among cultures and individuals, but share basic common themes. Privacy is sometimes related to anonymity, the wish to remain unnoticed or unidentified in the public realm.

Safety requirements

Safety can also be defined to be the control of recognized hazards to achieve an acceptable level of risk. Safety is the state of being "safe", the condition of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non- desirable.

OBJECTIVES OF SYSTEM

- It is very user-friendly and has added more features.

- To keep transaction details, customer details etc.

- Provide security to data - User Authorized system.

- To Reduce Clerical work and manpower.

- Decrease manual mistakes.

- Enable user’s workspace

- Customers can perform tasks through online mode.

SAMPLE PROGRAM CODE

DbConnect.java

```
package bank_management;

/*

@author MakeWealth

*/

import java.sql.*;

import javax.swing.JOptionPane;
```

```
public class Dbconnect {
static String url = "jdbc:mysql://localhost:3306/";
static String user = "root";
static String password = "";
public static Connection dbconnection(){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection(url,user,password);
return conn;
}

catch(Exception e){
JOptionPane.showMessageDialog(null,e);
}
return null;
}
```

Insert_data.java

```
package bank_management;
import java.sql. * ;
import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import javax.swing.JOptionPane;

/**

*

* @author MakeWealth

*/

public class insert_data {

Connection conn;

ResultSet res;

PreparedStatement pst;

Statement stt;

insert_data() {

try {

conn = Dbconnect.dbconnection();

stt = conn.createStatement();

stt.execute("CREATE DATABASE IF NOT EXISTS bank_management");

stt.execute("USE bank_management");

//user table and insert
```

```
stt.execute("CREATE TABLE I

NOT EXISTS user(" + "id INT not null," + "pass VARCHAR(20) not null," + "ac_type VARCHAR(15)," +
"join_date DATE," + "nation VARCHAR(15)," + "nid BIGINT not null," + "name VARCHAR(30)," + "p_number
VARCHAR(14)," + "dob DATE," + "address VARCHAR(50)," + "PRIMARY KEY(id)" + ")");

//money table and insert

stt.execute("CREATE TABLE IF NOT EXISTS money(" + "id VARCHAR(15) not null," + "name VARCHAR(30)
not null," + "amount BIGINT(12)," + "PRIMARY KEY(id)" + ")");



//history table and insert

stt.execute("CREATE TABLE IF NOT EXISTS history(" + "id VARCHAR(15) not null," + "amount BIGINT(12)
not null," + "date date not null," + "purpose VARCHAR(20) not null," + "t_to_id VARCHAR(20)" + ")");




//admin section   ................................//

stt.execute("CREATE TABLE IF NOT EXISTS admin(" + "id VARCHAR(15) not null," + "pass VARCHAR(20)
not null," + "name VARCHAR(30)," + "p_number VARCHAR(12)," + "PRIMARY KEY(id)" + ")");
```

```
//Executive section ....................................//

stt.execute("CREATE TABLE IF NOT EXISTS executive(" + "id VARCHAR(15) not null," + "pass
VARCHAR(15) not null," + "join_date DATE," + "nation VARCHAR(15)," + "nid BIGINT not null," +
"designation VARCHAR(25)," + "name VARCHAR(30)," + "p_number VARCHAR(14)," + "dob DATE," +
"address VARCHAR(50)," + "PRIMARY KEY(id)" + ")");

stt.close();

conn.close();

}

catch(Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

}
```

CONCLUSION

- The project Make Wealth Bank System is for computerizing the working in a bank.

- It is a great improvement over manual system. The computerization of the system as speed up the process
- The project Make Wealth Bank System was thoroughly checked and tested with dummy data and thus its found to be very reliable. The software takes care of all the requirements of an average bank and is capable to provide easy and effective storage of information related to Customer, executive, monitory transactions that come up for the bank service..

- It generates test reports and also provides the facilities for searching the details of the customer. It also provides report generation facility for all three admin, executive and customer on the basis of total balance or transaction.

LIMITATION OF THE SYSTEM

- Deals with only saving accounts.

- Since it is software, it needs to be manually installed on the banking staff desktop.

FUTURE ENHANCEMENT

- Staff login & signup window.

- Improvements in management and accountability.

- More security features, such as iris or facial recognition.

- Hashing algorithms to save hashed data.

BIBLIOGRAPHY

- Reference books from library Software Engineering By Roger Pressmen

- JAVA

- Website Referred https://www.programiz.com/java-programming

- Advice and Guidance by

  Prof. Leena Patil

  Prof. Milind Deshkar